This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1.   (Currently Amended) A method for input parameter binding, comprising:

at bind time, <u>calculating and </u>storing optimization information in a bind-in structure~~,~~ ~~wherein the bind-in structure has an associated section number~~;

when executing a statement <u>for one of a multiple row insert and a multiple row fetch,</u> when performing bind-in of host variables[[,]] <u>:</u>

<u>for column array processing for local processing,</u> comparing data in an application structure received with the statement with optimization information in the bind-in structure, wherein the optimization information includes at least one of data type, length, Coded Character Set Identifier, an array size, an indication of whether conversions are required, and an indication of whether the required conversions are valid, ~~wherein the statement has an associated section number,~~ wherein the application structure describes data, wherein the application structure is used to store data to be retrieved for a fetch statement, and wherein the application structure is used to provide data to be inserted for an insert statement; <u>and</u>

<u>for stream array processing for distributed processing, comparing the optimization information in the bind-in structure with information in a stream flow that includes metadata that is sent with the data;</u>

when there is a match ~~between the data in the application structure and data in the~~ ~~optimization information in the bind-in structure~~, executing the statement with the optimization information to perform one of fetching data from the data store and inserting data into the data store~~, wherein the bind-in structure and the statement have a same section number~~; and

when there is not a match ~~between the data in the application structure and the~~ ~~optimization information~~,

regenerating optimization information; and

executing the statement with the regenerated optimization information to perform one of fetching data from the data store and inserting data into the data store.

2.   (Cancelled)

3.   (Cancelled)

4.   (Cancelled)

5.   (Original) The method of claim 1, further comprising:

for fixed length data,

        storing an increment length by which a data pointer that is pointing to data in an

application program area is to be incremented to find a location of a next data value; and

        calculating the location of the next data value by adding the increment length to

the data pointer.

6.   (Original) The method of claim 1, further comprising:

for distributed processing, at a client computer, calculating a location of data in a client

communications buffer.

7.   (Original) The method of claim 1, further comprising:

for distributed processing, at a server computer, calculating a location of data in a server

communications buffer.

8.   (Original) The method of claim 1, further comprising:

for distributed processing, at a client computer, calculating a location of data in an

application program address space.

9.   (Original) The method of claim 1, further comprising:

when returning a handle to a cursor to a result set from a stored procedure to an

application, recalculating the optimization information.

10.   (Currently Amended) A method for output parameter binding, comprising:

at bind time, <u>calculating and</u> storing optimization information in a bind-out structure ~~wherein the bind-out structure has an associated section number~~;

when executing a statement <u>for one of a multiple row insert and a multiple row fetch,</u> when performing bind-out of host variables[[,]] <u>:</u>

<u>for column array processing for local processing,</u> comparing data in an application structure received with the statement with optimization information in the bind-out structure, wherein the optimization information includes at least one of data type, length, Coded Character Set Identifier, an array size, an indication of whether conversions are required, and an indication of whether the required conversions are valid, ~~wherein the statement has an associated section number,~~ wherein the application structure describes data, wherein the application structure is used to store data to be retrieved for a fetch statement, and wherein the application structure is used to provide data to be inserted for an insert statement; <u>and</u>

<u>for stream array processing for distributed processing, comparing the optimization information in the bind-out structure with information in a stream flow that includes metadata that is sent with the data;</u>

when there is a match ~~between the data in the application structure and data in the optimization information in the bind-out structure~~, executing the statement with the optimization information to perform one of fetching data from the data store and inserting data into the data store~~, wherein the bind-out structure and the statement have a same section number~~; and

when there is not a match ~~between the data in the application structure and the optimization information~~,

regenerating optimization information; and executing the statement with the regenerated optimization information to perform one of fetching data from the data store and inserting data into the data store.

11.   (Cancelled)

12.   (Cancelled)

13.   (Cancelled)

14.   (Original) The method of claim 10, further comprising:

for fixed length data,

storing an increment length by which a data pointer that is pointing to data in an application program area is to be incremented to find a location of a next data value; and

calculating the location of the next data value by adding the increment length to the data pointer.

15.   (Original) The method of claim 10, further comprising:

for distributed processing, at a client computer, calculating a location of data in a client communications buffer.

16.   (Original) The method of claim 10, further comprising:

for distributed processing, at a server computer, calculating a location of data in a server communications buffer.

17.   (Original) The method of claim 10, further comprising:

for distributed processing, at a client computer, calculating a location of data in an application program address space.

18.   (Original) The method of claim 10, further comprising:

when returning a handle to a cursor to a result set from a stored procedure to an application, recalculating the optimization information.

19-38. (Cancelled)

39.   (New) The method of claim 1, wherein the statement is for a single row fetch.

40.   (New) The method of claim 10, wherein the statement is for a single row fetch.